AD NUMBER

ADB216713

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

Distribution authorized to DoD only; Specific Authority; . Other requests shall be referred to Commander, U.S. Army Medical Research and Materiel Command, Attn: MCMR-RMI-S, Fort Detrick, Frederick, MD 21702-5012.

AUTHORITY

USAMRMC ltr, 4 Dec 2002

AD______________

CONTRACT NUMBER DAMD17-96-C-6021

TITLE: Medical Decision Algorithm for Pre-Hospital Trauma Care

PRINCIPAL INVESTIGATOR: Donald K. Wedding, P.E., Ph.D

CONTRACTING ORGANIZATION: Photonics Systems, Incorporated
Northwood, Ohio 43619

REPORT DATE: September 1996

TYPE OF REPORT: Final, Phase I

19961202 030

PREPARED FOR: Commander
U.S. Army Medical Research and Materiel Command
Fort Detrick, Frederick, Maryland 21702-5012

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 1996

3. REPORT TYPE AND DATES COVERED
Final, Phase I (15 Mar 96 - 14 Sep 96)

4. TITLE AND SUBTITLE
Medical Decision Algorithm for Pre-Hospital Trauma Care

5. FUNDING NUMBERS
DAMD17-96-C-6021

6. AUTHOR(S)
Donald K. Wedding, P.E., Ph.D.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Photonics Systems, Incorporated
Northwood, Ohio 43619

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Commander
U.S. Army Medical Research and Materiel Command
Fort Detrick, Frederick, MD 21702-5012

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

4 DEC 1996

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution authorized to DOD Components only (specific authority). Other requests for this document shall be referred to Commander, U.S. Army Medical Research and Materiel Command, ATTN: MCMR-RMI-S, Fort Detrick, Frederick MD 21702-5012.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The scope of work in this Phase I research centered on three areas: 1) data acquisition, 2) neural network design, and 3) system architechture design.

In the first area of this research, a triage database was located containing about 175,000 patients. This data will be used in the Phase II research.

The second area of this research was a designs and prototype of a neural network which was able to successfully identify the triage status of patients over 90% of the time. It was able to do this in under 1 second. This prototype will be expanded upon and improved in Phase II.

The third area focused on generating an overall program architecture which will allow for additional decision support systems to be added over time, such as as diagnoses and treatment recomendations, and sensor additions and malfunctions.

14. SUBJECT TERMS
triage, decision support, medical decision system, diagnosis expert system, neural network, non-invasive sensor

15. NUMBER OF PAGES
24

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Limited

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C -	Contract	**PR** -	Project
G -	Grant	**TA** -	Task
PE -	Program Element	**WU** -	Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

Attachment J.5.

FOREWORD

Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the U.S. Army.

( ) Where copyrighted material is quoted, permission has been obtained to use such material.

( ) Where material from documents designated for limited distribution is quoted, permission has been obtained to use the material.

( ) Citations of commercial organizations and trade names in this report do not constitute an official Department of the Army endorsement or approval of the products or services of these organizations.

( ) In conducting research using animals, the investigator(s) adhered to the "Guide for the Care and Use of Laboratory Animals," prepared by the Committee on Care and Use of Laboratory Animals of the Institute of Laboratory Animal Resources, National Research Council (NIH Publication No. 86-23, Revised 1985).

( ) For the protection of human subjects, the investigator(s) have adhered to policies of applicable Federal Law 32 CFR 219 and 45 CFR 46.

( ) In conducting research utilizing recombinant DNA technology,the investigator(s) adhered to current guidelines promulgated by the National Institutes of Health.

LD-KW-P, P.E., Ph.D. — Sept. 11, 96

Principal Investigator's Signature — Date

TABLE OF CONTENTS

INTRODUCTION

The work for Phase I had three primary focuses: 1) triage data acquisition, 2) decision support system design and prototype, and 3) design of an overall architecture to allow updating and expansion of the systems' functions.

1) In the first area, an analysis was conducted as to existing databases in the field of triage. This database would be used to train and test the decision support systems in Phase II. It was believed by the Principal Investigator that locating a source of data was the most important feature of Phase I research because without data, training a decision support system would be virtually impossible.

   The database acquisition part of this research was a great success as the Principal Investigator was put in contact with two individuals who possess a triage database containing approximately 175,000 entries. The owners of this database have agreed to work as consultants for Phase II of this research.

2) The second focus of this research was to prototype a decision support system. Although there are numerous methodologies in decision support science such as statistical analysis, fuzzy logic, and data clustering, the methodology selected for this application was a Radial Basis Function (RBF) Neural Network.

   An RBF neural network was prototyped to accept physiological sensor input data and classify a patient as RED, YELLOW, or GREEN depending upon the seriousness of the injuries. The neural network was able to perform this function in microseconds which was considerably less than the 15 second requirement in the SBIR Phase I announcement.

3) The third focus was a structural design of a program which would allow the triage system to expand and adapt over time. As new data becomes available and new needs are determined, such as injury diagnosis and treatment recommendations, the decision support system will be adaptable to these new requirements.

   In addition, the program structure designed in Phase I will allow for the condition of sensors being damaged. In other words, if one of the non-invasive sensors monitoring a soldier is damaged, the system will adapt by shutting down the decision support systems which require that sensor input and activating alternate decision support systems not requiring that sensor input. This is far more desirable than having the whole system collapse because of unavailable data.

BODY

DATA ACQUISITION:

There is one common need to any decision support system, regardless of its design methodology, neural network, fuzzy logic, pattern recognition, or statistical regression. That common need is data.

Without data, the system can be neither trained nor tested. Therefore, a large supply of accurate data is necessary. The search for data was one of the most critical issues of the Phase I research.

This part of the research was successful far beyond the expectation of the Principal Investigator. The Principal Investigator was given the name of two researchers in the area of triage by Major Dr. Stephen P. Bruttig. The two researchers were Dr. Bill Sacco and Dr. Howard Champion.

These two researchers manage the Major Trauma Outcome Study (MTOS) and Pennsylvania Trauma Outcome Study (PTOS) databases. The MTOS database consists of approximately 175,000 blunt-injured and penetrating-injured trauma patients treated at 166 hospitals during 1982-1989. The MTOS patients were Level I or Level II trauma centers whose submissions account for more than 95% of the database. Not only do these two individuals possess the necessary data required for a decision support system, they are also recognized experts in the area of trauma medicine.

The Principal Investigator is pleased to announce that these individuals have agreed to act as consultants on the Phase II part of this research. As a result, both their database and personal expertise will be available and highly utilized in the continuation of this research in Phase II.

DECISION SUPPORT SYSTEM:

Programming Language Selection

Ada was selected as the language to be used for this neural network for three reasons: 1) it is the military standard, and 2) it is the safest language for a medical application, and 3) it is the most portable of all languages.

The first consideration was that the two standards in force, DOD-STD-2167A and DOD-STD-2168 require special permission to NOT use Ada. Although obtaining permission to avoid Ada is not difficult, the existence of the standards above indicate that the Federal Government and in particular the Military prefer Ada over other programming languages. Because the primary final user of this software will be the United States Army, and possibly

other branches of the military, it was believed that the use of Ada would make it more likely that the software developed in this project would be incorporated into other related projects.

Another benefit of Ada is that it has very strict and rigid requirements which will flag such activities as misassigning variables and arrays being indexed out of bounds. This was a very important factor in the final consideration since neither C/C++ nor Fortran have these constraints. Thus, using either C/C++ or Fortran may result in very subtle bugs being missed which would not be manifested until the software was deployed. In a medical application such as this, such risks were determined unacceptable. Ada was therefore considered to be the safest choice.

The third consideration for using Ada is that the rigid ANSI standard of the language makes it portable to other platforms. Since the final platform which will run this software has not yet been determined, Ada software will have the greatest chance of running without any major software revisions. Although C/C++ and Fortran both have ANSI standards, these standards were only written long after the languages. Thus, there are many dialects of these languages in existence. Transporting C/C++ and Fortran from one compiler to another or one platform to another generally requires tremendous amounts of software revision. Ada is therefore the most portable of the languages.

RBF Neural Network Design

The decision support methodology that was used for triage determination was the RBF neural network. In order to maximize portability and reusability of code, ada generic packages were used in the implementation of this program.

The center selection routines were written and debugged. There was time to code two different methods, the Node At Data Point approach and the Hard C Means Clustering Algorithm. Both approaches are widely used in Radial Basis Function (RBF) neural networks and tend to yield good results. Two different methods for radius selection were also coded, the constant radius and the Alpha Nearest Neighbor method. The code for calculating the RBF Neural Network weights was written and debugged. The network can now be trained on virtually any data set.

The result of this effort was Ada source code which will be used in Phase II of this research for determining triage classification. The RBF network was then tested which is described below.

Several variations of the certainty factors were coded into the RBF neural network. The certainty factors were then used in the training of the network.

The neural network was trained using the triage data set generated by the information from Major Bruttig and the rules from an ER Physician. The network using the certainty factor

architecture (described below) was trained to identify hypothetical patients as RED, YELLOW, or GREEN. The network was tested on 700 data points. Of these 700, 654 were diagnosed and the remaining 46 had low certainty factors and were not used. Of the 654 points, 599 were successfully diagnosed for an accuracy of almost 92%. Also virtually all of the patients that were misclassified were patients who were on the borderline between RED and YELLOW and the borderline between YELLOW and GREEN. These borderline cases could have gone either way. With a more complete dataset and more training time, it is believed that this 92% could be significantly improved.

Note that the time it took to calculate the 654 data points was less than 1 second. This indicates that the network is certainly more than capable of diagnosing a single data point within the 15 second requirements. It appears that the network has over 14 seconds to spare.

The RBF certainty factor architecture employed for this neural network is shown in Figure 1. In this architecture, the RBF Neural Network Generates both an output and a certainty factor. The certainty factor can assume a value of -1.0 to 1.0. A certainty factor of -1.0 indicates that the network believes that there is a 0% probability that the result is not correct, while a value of 1.0 indicates that the network is 100% confidant in the result.

The method for calculating certainty factors used in this neural network prototype is an improvement on the variation in the proposal. The method used in this prototype uses the actual output for all of the possible classifications to generate a certainty factor. For example, assume that the network's output for a RED classification is 0.3, YELLOW is 0.8, and GREEN is 0.2. The classification is given as YELLOW since that has the greatest output. Now the certainty factor is given as follows, since the output values have an approximate range of 0.0 to 1.0, the values can all be treated as certainty factors.

Therefore, the network is 30% certain the patient is RED, 80% certain that the patient is YELLOW, and 20% certain that the patient is GREEN. It can also be assumed that the RED and GREEN certainties indicate a certainty that the output is NOT YELLOW, or certainties of -30% and -20% respectively. Two negative certainties are combined using the formula:

$$CF_{ab} = CF_a + CF_b + (CF_a * CF_b) \tag{1}$$

In Equation (1), the certainty factors for the evidence *a* and *b* are combined to form a new certainty. If there are more than two pieces of evidence, for example *a*, *b*, and *c*, they can be combined using Equation (1). This can be accomplished by first merging CF_a and CF_b to yield CF_{ab}. This new certainty factor is then combined with CF_c to form CF_{abc}. It is important to note that combining rules with Equation (1) is not order dependent. In other words, CF_a and CF_c can be combined first to form CF_{ac}. CF_b can then be merged with CF_{ac} to yield CF_{acb}. This result would be the same as the CF_{abc} which was calculated above.

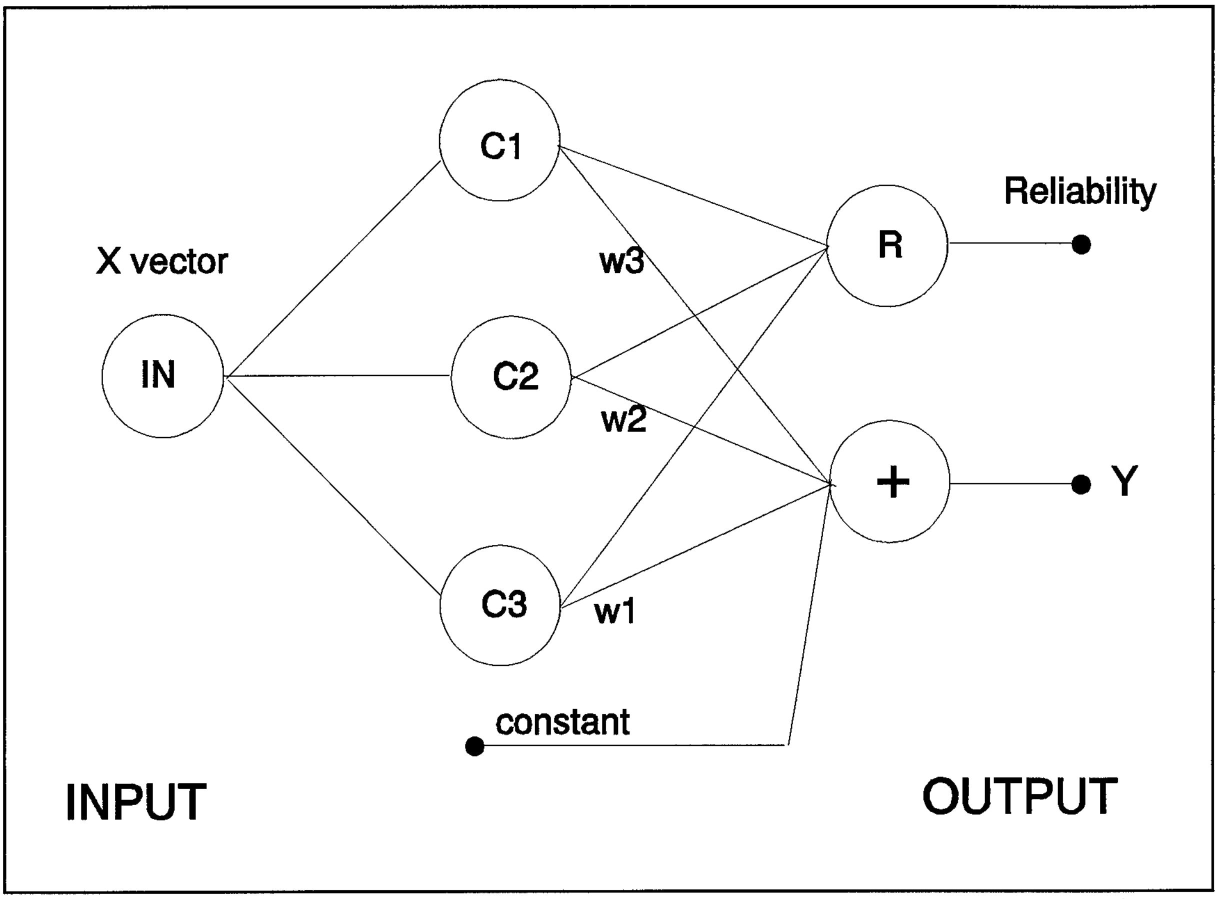

Figure 1 RBF Neural Network Architecture to Generate Both Output and Certainty Factors

For the example above, combining the RED and GREEN certainties together result in a certainty of -0.44 or -44%. The combining positive and negative certainty factors is accomplished with Equation (2) which is given as:

$$CF_{ab} = \frac{(CF_a + CF_b)}{1 - MIN(|CF_a|, |CF_b|)} \tag{2}$$

In Equation (2), the positive and negative certainty factors are added together to form the numerator. The denominator is the minimum of the absolute values of the two certainty factors. So, combining the 80% certainty that the output is yellow with the 44% certainty that it is not YELLOW results in an overall certainty of 64%.

The certainty factor architecture was incorporated in the following manner. An arbitrary cutoff point was selected. In the prototype discussed above, the cutoff was 30%. Any output that is above 30% is displayed, while any output that is below 30% is considered unfamiliar

to the network. This unfamiliar data is then stored internally in the computer where it can be analyzed by an expert in the medical field as to the correct diagnosis. The architecture is seen in Figure 2.

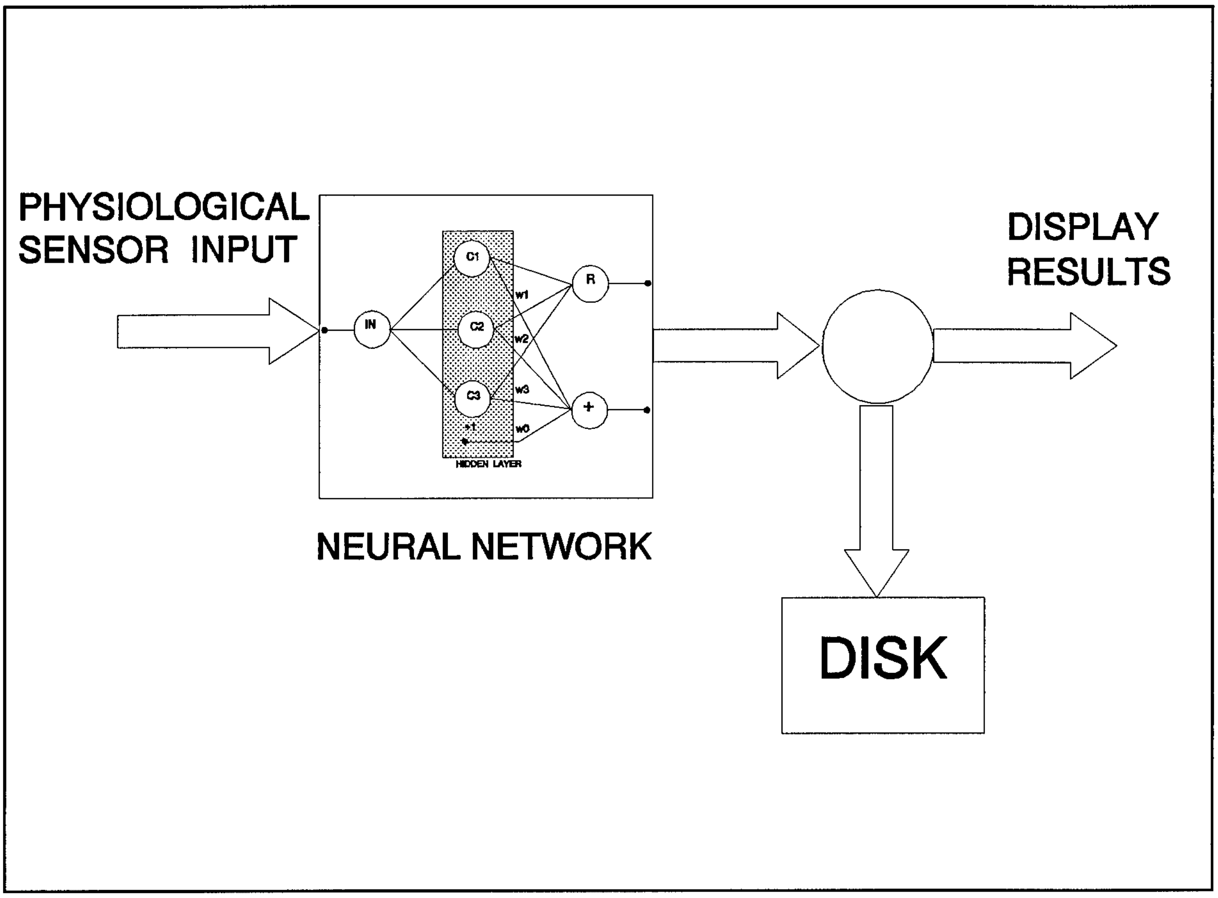

Figure 2 Medical Decision Systems Prototype

The unfamiliar data can then be used to either retrain the current network or else train a fallback network cascaded onto the first. Thus, when input enters the network and output and certainty are determined, if the certainty is below cutoff, then the input is fed into the fallback network. If the certainty is again too low, it is fed into yet another network. This process continues until the system either achieves a satisfactory result or it runs out of networks in which case the output is saved in memory.

This flagging of unfamiliar data is an advantageous feature since every decision support system will eventually encounter unfamiliar data; however, RBF networks will know that the data is unfamiliar and flag it as such. It can then be retrained to recognize it in the future. The RBF network, therefore, will grow more accurate over time.

The *ada generic* neural network approach has been tested. The architecture allows for initial training of the data. As data is entered into the decision support system calculated output diagnosis. Unfamiliar data, was also sent to a file for further analysis.

The unfamiliar data was then used to train a new neural network with the same code used in the first neural network. This reusability of code will prove to be cost effective over time for the military. The new network then works as follows:

> Data enters the first network, it calculates an output and a reliability. If the reliability is high, then the systems stops. If the reliability is low, then it proceeds to the new network.
>
> The new network follow the exact same function, only this time if the data is unfamiliar then it is written to a file where an expert can examine it later. When enough data is collected, a third network will be trained.
>
> This architecture has been tested and found to work. It is important to note that this decision support system, like all other decision support systems, will not be perfectly accurate. However, this architecture will have the ability to flag unfamiliar data which can be used to train new networks to supplement the original.

Therefore, over time this network will become increasingly more accurate.

<u>Graphic User Interface</u>
In addition to the RBF neural network, a Graphic User Interface (GUI) was merged with an ada source program. The GUI interface allowed the user to enter data from a menu and simulate non working sensors. A full scale GUI interface will be used in Phase II as a testing and debugging tool of the expert system. The GUI interface, however, will be detachable from the actual expert systems code, so that the actual code can be burned into a microchip and placed on the soldier's portable computer.

PROGRAM STRUCTURE:

<u>Algorithm Design</u>

The following is a software architecture which will accept input from non-invasive sensors and output medical diagnosis from this input. The architecture was developed during Phase I of this research.

The architecture has three main components: 1) Non-Invasive Sensor Routines 2) Main Program 3) Decision Support System. These three components will be described in detail below along with a description of how the components will fit together. In addition to the algorithm design, a menu interface will be developed to enable testing, training, simulation, and debugging of the program.

Non-Invasive Sensor Routines

Each non-invasive sensor will be continuously monitored by a background computer program such as a Terminate Stay Resident (TSR) program. As the algorithm is designed to be written in the Ada programming language, Ada *task* structures will be used for implementation of these background programs. Ada tasks are defined in the programming language and allow for procedures to run independently of a Main Program. The block diagram of the non-invasive sensor programs is given in Figure 3.

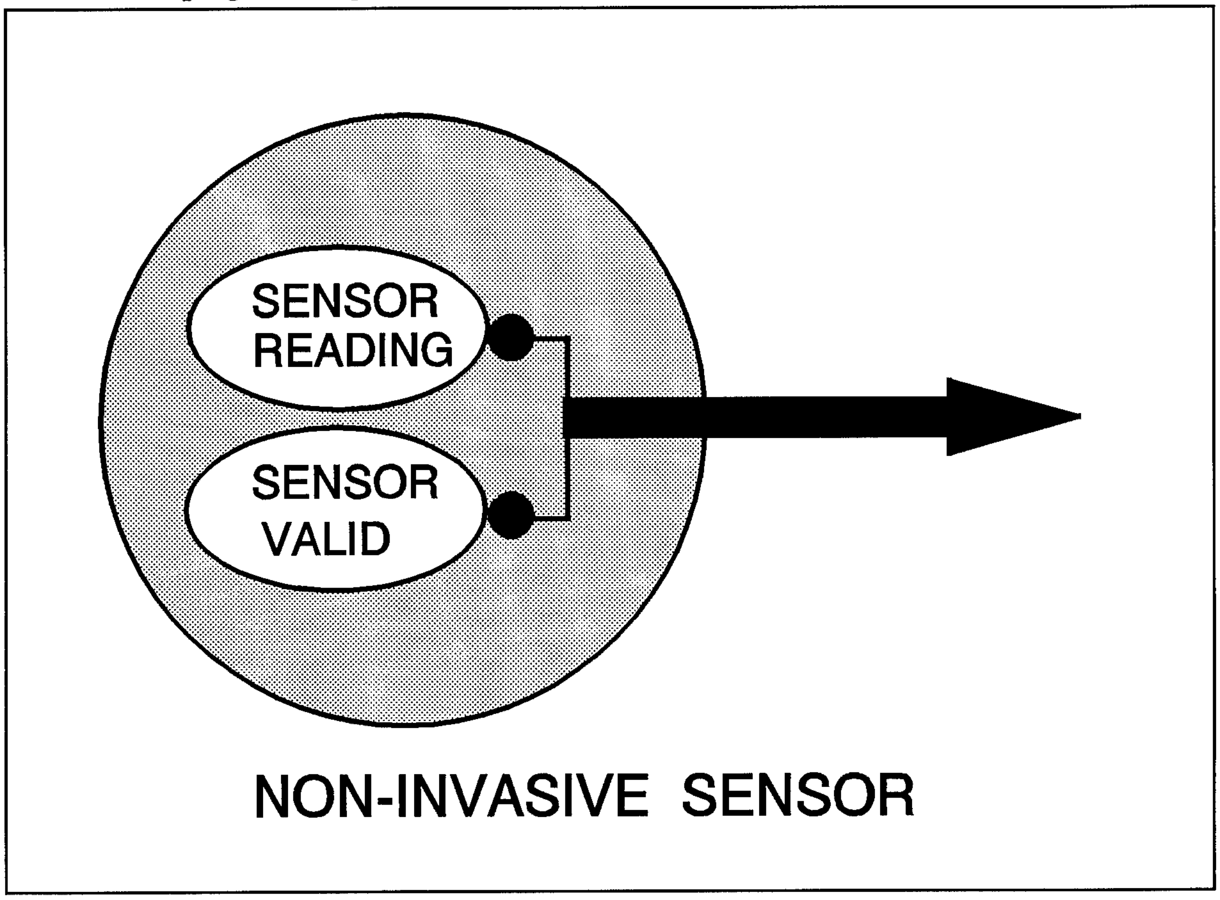

Figure 3

Each Ada task will continuously monitor a specific non-invasive sensors. The Ada task will, at any given time, contain the most recent reading from that sensor. In addition, the Ada task will be responsible for determining if the sensor is or is not still functioning. This *validity flag* will account for the possibility that the sensor in question is no longer functioning. This condition might occur when a bullet wounding a soldier also damages the sensor.

As noted above, the Ada tasks will be running *concurrently* with one another and with the Main Program. When the Main Program requires updated information, it will request the latest readings from each of the sensors. Each sensor, upon being asked, will provide the

Main Program with its latest reading along with a flag indicating whether or not the reading is reliable. The Main Program will know whether the information should or should not be used when formulating medical information.

Body Temperature Sensor Example:

Assume an Ada task is written to monitor a soldier's body temperature. The following transactions might occur:

- The Ada task reads the sensor at **37.0°C**, the reading is **Valid**.
- The Ada task reads the sensor at **36.9°C**, the reading is **Valid**.
- The Ada task reads the sensor at **37.1°C**, the reading is **Valid**.
- The Main Program polls the Ada task. The Ada task sends the value **37.1°C** and a flag indicating the reading is **Valid**.
- The Ada task reads the sensor at **37.0°C**, the reading is **Valid**.
- The Ada task reads the sensor at **37.1°C**, the reading is **Valid**.
- The sensor is damaged and begins giving faulty data.
- The Ada task reads the sensor at **44.0°C**, the reading is **Invalid**.
- The Ada task reads the sensor at **22.0°C**, the reading is **Invalid**.
- The Main Program polls the Ada task. The Ada task sends the value **22.0°C** and a flag indicating the reading is **Invalid**.
- The Ada task reads the sensor at **26.0°C**, the reading is **Invalid**.
- The soldier notices the damaged sensor and is able to repair it.
- The Ada task reads the sensor at **37.1°C**, the reading is **Valid**.
- The Ada task reads the sensor at **37.0°C**, the reading is **Valid**.
- The Main Program polls the Ada task. The Ada task sends the value **37.0°C** and a flag indicating the reading is **Valid**.

It is therefore shown that the processess of reading the sensors with Ada tasks will be continuous and independent of the Main Program. The Ada task is only concerned with reading the sensor and determining whether or not the sensor is functioning.

It will supply the data to the Main Program only when requested. This will eliminate the problem of the Main Program being slowed down by receiving unneeded data. Also, if there is a delay by the Ada task in reading new data, the Ada task will simply supply the most recent data available to the Main Program. The Main Program will, therefore, never be forced to wait on the Ada task.

In addition, the Ada task will alert the Main Program when data is invalid. The Main Program will then know to discount this information. The Ada task will also be able to detect if the sensor suddenly becomes functional again, through repair, for example.

The advantage of using Ada tasks over standard procedures is that they will be able to monitor their sensors at their own speed, and will not cause the Main Program to slow to the

speed of the slowest sensor. In addition, if the sensor is damaged and no data is available, the Main Program will not "hang" while waiting for data which will never arrive. The Ada task will provide the Main Program with its most recent data, even if the sensor has not yet updated the reading.

In other words, the use of concurrent programming techniques will ensure that the Main Program will run at its fastest possible speed and will not crash or halt due to failed sensors.

From discussions with consultants, some possible sensor values will be considered in the Phase II research. These sensors and their uses will be discussed below.

Standard Medical Sensors:

The first set of sensors employed will be the standard, non-invasive medical sensors described in the Phase I offer. These sensors may include such medical readings as, body temperature, heart rate, tissue pH, cardiac output, and respiration rate. The actual medical sensors employed will be determined at the outset of the Phase II research from the available database and existing technology.

Dual Use Sensors:

Nonmedical sensors can also serve a secondary function in medical diagnosis. The Global Positioning System (GPS) carried by the soldier, for example, can also provide information to the diagnositic system. It can be used to provide information as to whether the soldier is capable of movement. This information can be employed in the triage classification.

It is important to note, however, that if the GPS system indicates the soldier is not moving, it does not necessarily mean that the soldier is incapable of movement. Therefore, this type of dual use sensor can only be used in specialized circumstances. The Main Program will use the GPS (or other dual use sensor) when available and when it indicates motion, otherwise, the Main Program will rely strictly on the medical sensors.

Active Sensors:

In addition to passive sensors such as medial readings and global positioning, the soldier can also be outfitted with active sensors. Active sensors can provide additional and more accurate information than passive sensors. The drawback to the use of active sensors is that their information is not always available.

One possible active sensor might allow a medic to enter simple data about a patient that is not discernable from the passive sensors. These might include such information as the patient is conscious and able to speak. This information, if available, would be

invaluable in determining if a patient's triage status is RED, YELLOW, or GREEN. It could also be used in other simple diagnosis of the patient.

Other possible active sensors might include a vibration unit as is found on most home "pagers". The vibration unit could be sent a signal from a distance causing it to vibrate briefly. The injured soldier could then acknowledge the vibration unit, which would indicate that the soldier was alert and able to follow simple commands. This again would be invaluable to determining the patients survivability chances.

In summary, active sensors can be employed to increase accuracy, but if the active input is unavailable, the medical decision support system will be able to rely strictly on passive information, such as heart rate and body temperature, for diagnosis.

Main Program

The block diagram for the Main Program is given in Figure 4. This procedure is charged with accepting input from the non-invasive sensors and sending the correct data to the appropriate Decision Support Systems.

The flow of the Main Program is described as follows. First, the Main Program polls all of the Ada tasks which monitor the sensors. It receives from each Ada task its most recent value and a validity flag.

Although the Ada tasks should already have flagged "bad data", there is a possibility that there will still remain an unflagged malfunction. The Main Program, will therefore, run a final check on the incoming data to determine if it is reasonable.

For example, assume the Ada task monitoring body temperature sends a value of **250.0°C** and also a validity flag value of **VALID**. Obviously, this data is inaccurate and should not be used in medical calculations. Normally, this will be flagged by the Ada task as **INVALID**; however, given that technical malfunctions can occur, a secondary check will done by the Main Program.

The secondary check will not be as exhaustive as the checks done in each of the Ada tasks, but will provide enough redundancy that no obvious errors will slip through which might harm the patient. The Main Program in the above example would run a check to determine if the temperature is within a reasonable range, say 25.0°C to 50.0°C. If the data point falls outside of this range, then the data would be determined to be **INVALID** regardless of the validity flag sent by the Ada task.

The sensor readings and their validity are then sent to all of the Decision Support Systems (DSS) within the Main Program.

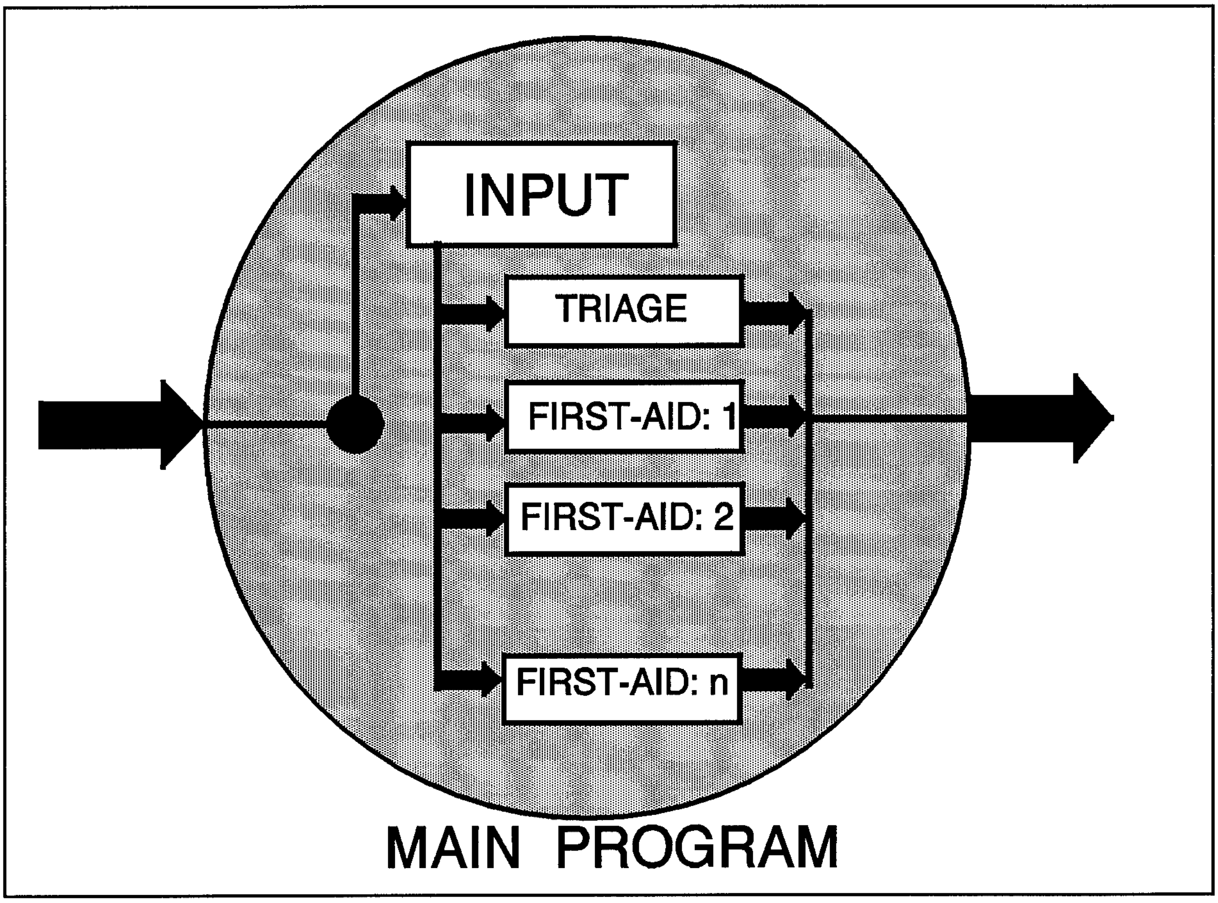

Figure 4

Note that at this time, the Army is only investigating the use of a DSS for use in triage evaluation; however, the Army has expressed an interest in expanding this application in the future to accomplish such tasks as simple diagnosis and suggested treatment for simple, albeit life threatening, injuries. This architecture will allow for new DSS programs to be added to the application as they are developed, and allow obsolete DSS programs to be removed.

After each DSS has calculated an output for its own specific application, the Main Program sends the answer to the output device which may be anything from a GREEN, RED, or YELLOW Light Emitting Diode (LED) to a small Liquid Crystal Display.

<u>Decision Support System</u>

A block diagram for a hypothetical Triage Decision Support System (DSS) is given in Figure 5. From this diagram, it is seen that each DSS will be designed to be run by an Executive procdure and will contain one or more actual Decision Support System sub-programs (SUBDSS) within the DSS.

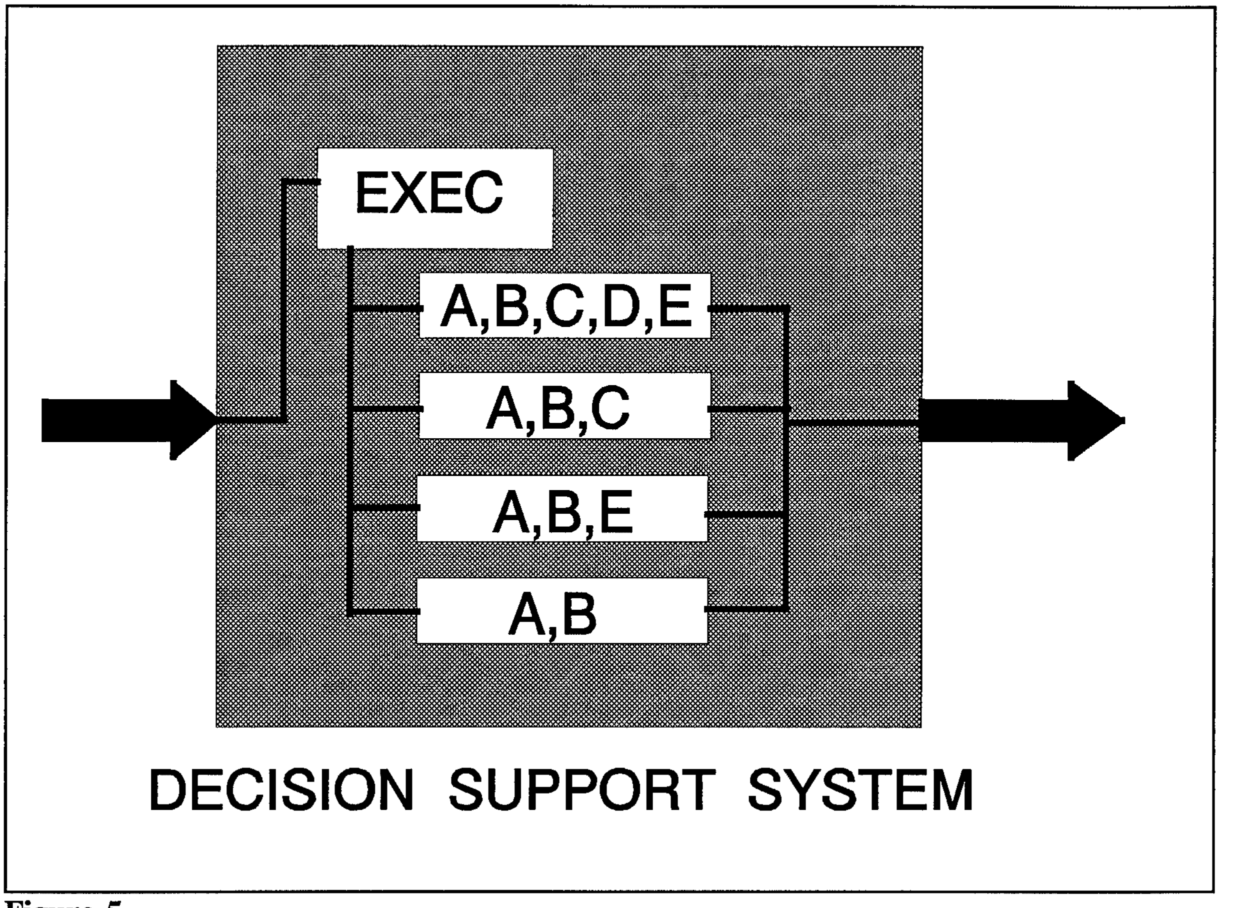

Figure 5

The executive procedure will receive the sensor readings and validity flags from the main program. From this information, it will execute the most accurate SUBDSS algorithm that it is capable of running.

In this hypothetical example, the Triage DSS is sent data from sensors A,B,C,D, and E. If all of this information is available and valid, the executive activates the top level SUBDSS, which will be the most accurate.

However if the output from sensor D is invalid, then the executive will activate a fallback SUBDSS. In this case, the executive will activate the second SUBDSS as sensors A,B, and C are still valid. If sensors C and E also malfunction, then the final SUBDSS is employed.

Note that this example requires that in all cases sensors A and B must function. If either of these malfunction, then no diagnosis is possible. Although it is certainly possible for the executive procedure to "guess" in this situation, this is not desireable. Certain data will be critical to a DSS. If the critical data is invalid then the DSS should not be used, as the results would be meaningless and in a medical application, dangerous.

There are three advantages to this DSS design: 1) it accounts for sensor malfunctions, 2) it allows for differing methodologies, and 3) it is expandable. These advantages are discussed below.

1) It is impossible to assume that the non-invasive sensors will always function. The same bullet that wounds the soldier can also damage the sensor. Rather than simply "guess" at the last value of the sensor or not make a diagnosis at all, this architecture allows the DSS to use the existing data available to arrive at a diagnosis. Even though this diagnosis will likely not be as accurate as a diagnosis with more inputs, it still will provide useful information to the individual administering medical assistance.

2) This architecture also allows for different types of SUBDSS methodologies. In the field of Decision Support Systems, there are numerous approaches employed including neural networks, fuzzy logics, pattern recognition, and statistical analysis. Although proponents of each will claim one approach is always the better, in actuality the best approach varies from application to application and with the data available.

   In the above example, the best approach with sensor readings A,B,C,D, and E might be a neural network while the SUBDSS requiring sensor readings A,B, and C might best be implemented with statistical analysis. This architecture will allow for whatever SUBDSS is most effective to be inserted into the architecture at a position indicating its effectiveness relative to the other SUBDSS routines.

3) The final advantage to this approach is that it allows for expansion. As new sensors are developed and new SUBDSS algorithms for the application are developed, they can be added. For example, consider the above Triage DSS example. If a new Triage SUBDSS is developed utilizing sensors C and E and it is more accurate than the SUBDSS utilizing A,B,C then this new SUBDSS will be inserted after the first and before the second SUBDSS in Figure 5.

This architecture will allow the Army to contract for new DSS and new SUBDSS routines to perform specialized tasks. Companies around the United States which possess specialized knowledge in one area can develop pieces for this architecture on a competitive basis. Upon the completion of the new DSS or SUBDSS, it can be inserted into the architecture.

Overview of Algorithm

An overview of the Algorithm is given in Figure 6. This architecture allows for a Main Program to receive data from any number of sensors.

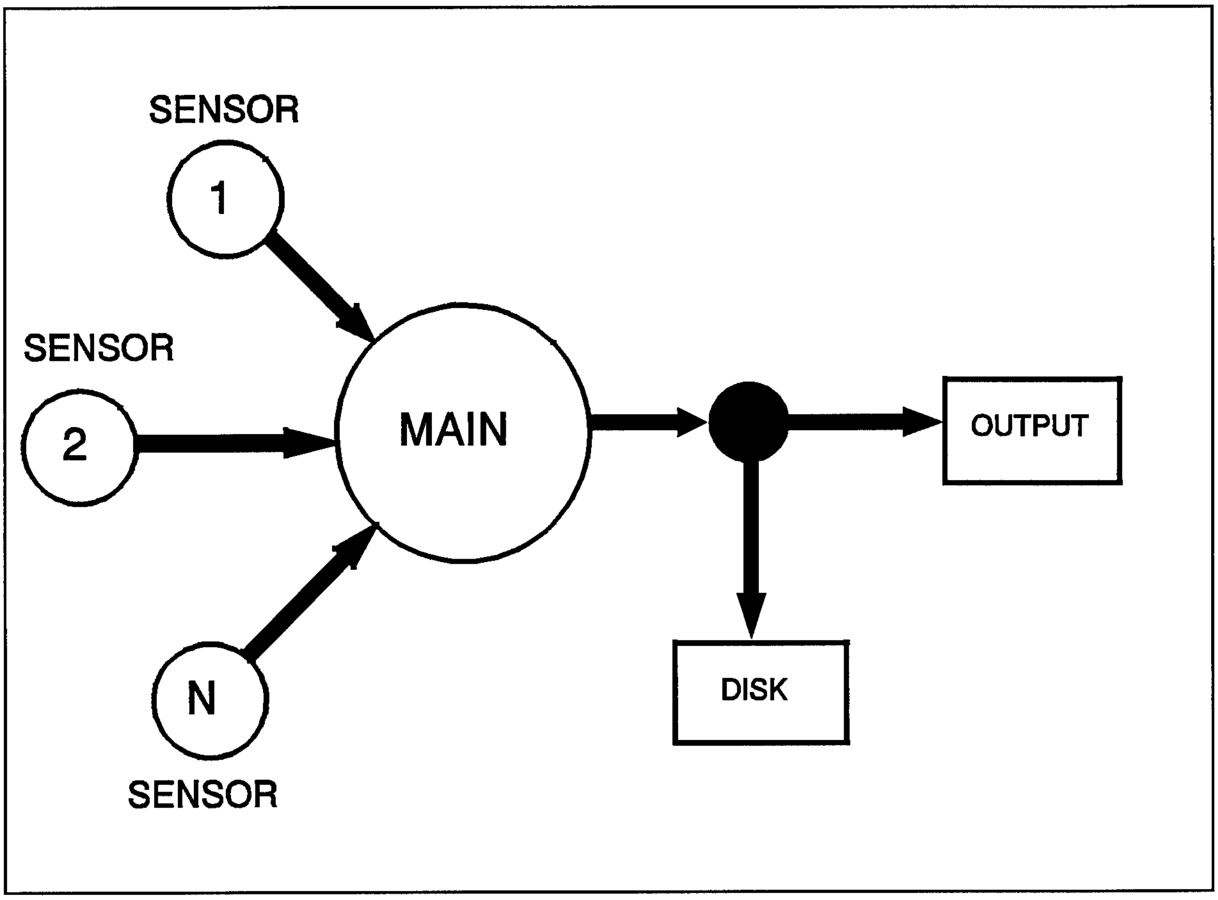

Figure 6

From this data, the Main Program activate the Decision Support System (DSS) routines such as classifying a patient's triage status as RED, YELLOW, or GREEN.

The output will be sent to an output device which may include Light Emitting Diodes (LED's) or small Liquid Crystal Displays (LCD's). In addition, the input data and output from the system will be saved in internal memory or on some storage device. This will allow for checking the accuracy of the system by examining the programs' outputs against those of actual experts. Saving the program's data will enable it to be updated and improved over time.

The proposed architecture of Figure 6 will also allow for any number and/or type of sensor inputs. Therefore, as new sensor types are developed, tissue pH for example, they can be implemented into the system with little trouble. This program design will, therefore, expand and grow with the needs of the end user.

Note that this program architecture is "stand alone" and carries no bulky user interface. Once it is deemed acceptable, it can be put into binary format and burned into a Programmable Read Only Memory (PROM) Chip.

Algorithm Graphic User Interface

The above architecture can be tested through a Graphic User Interface (GUI) shown in Figure 7. The GUI Menu will allow the developers of the program to test the DSS and SUBDSS programs as they are developed.

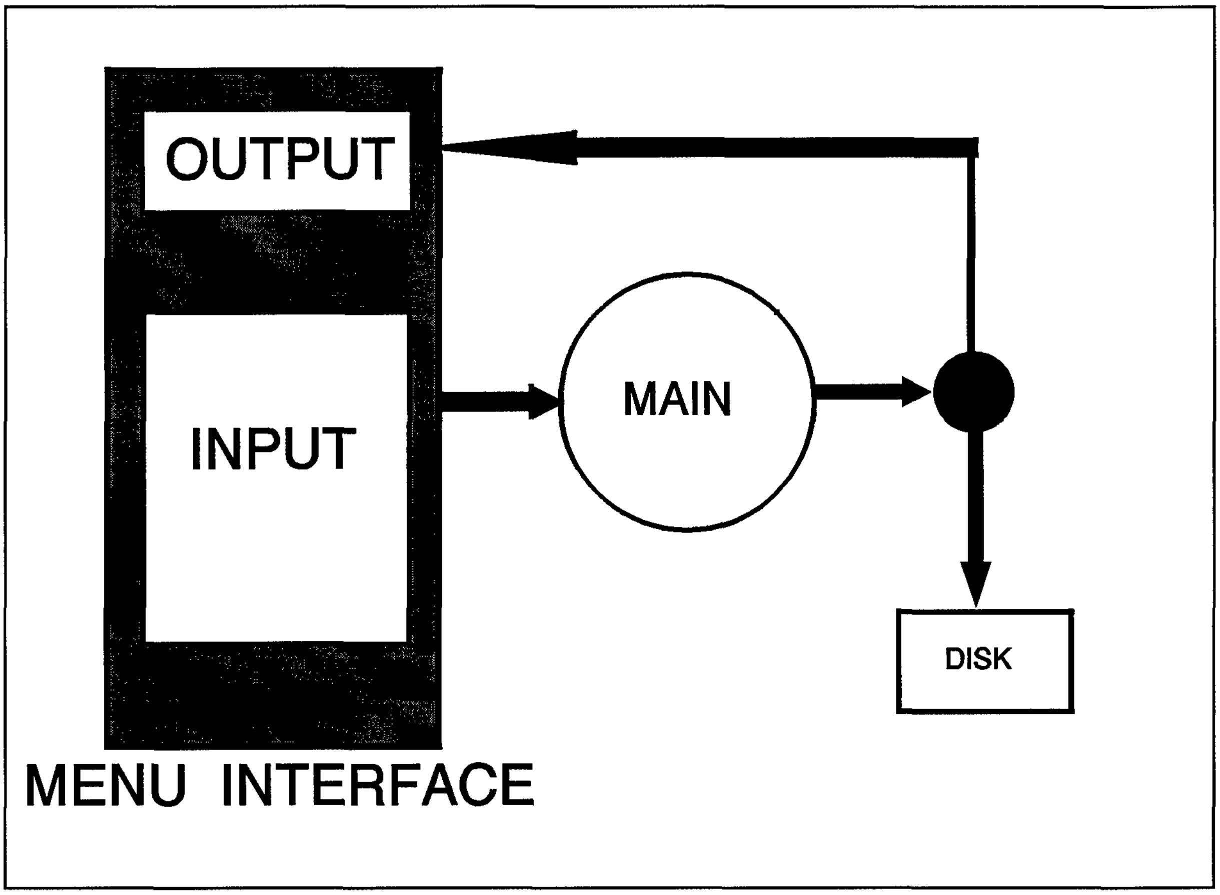

Figure 7

The GUI Menu will enable the user to enter data and set validity flags, which simulate the sensor input. The actual Main Program will then read these values the same way that it will read the Ada task values in the actual program. The advantage of this Menu Interface is that it can be used not only for testing and debugging, but for training and simulation as well.

Once the DSS and/or SUBDSS routines are tested and debugged using the GUI Menu, it can be detached leaving only the Main Program. The Main Program will then be converted into binary and burned into a PROM.

CONCLUSIONS

There were three goals outlined in the Introduction of this Phase I document: 1) acquisition of a data source, 2) design of a decision support system, and 3) design of a main program architecture. All three of these goals were met.

A data source was located along with the expertise to use this data when the consulting services of Dr. William Sacco and Dr. Howard Champion were secured. Dr. Sacco and Dr. Champion have access to the Major Trauma Outcome Study (MTOS) and Pennsylvania Trauma Outcome Study (PTOS) databases. The MTOS database consists of approximately 175,000 blunt-injured and penetrating-injured trauma patients treated at 166 hospitals during 1982-1989. The MTOS patients were Level I or Level II trauma centers whos submissions account for more than 95% of the database. Therefore, the necessary data will be available for the training of the decision support system.

Secondly, a decision support system based on RBF Neural Networks was coded and tested using hypothetical data supplied by an emergency room physician. The decision support system successfully identified approximately 92% of the patients. The time required to diagnose all 654 patients tested totalled significantly less than 1 second. The time required by the specification was that one patient must be diagnosed in under 15 seconds. Therefore, the time requirements for this decision support system were easily met. In addition, the RBF architecture allows for the flagging of unfamiliar input data. This data is saved for further analysis by triage experts. This data will be used to update the RBF network. Therefore, unlike other decision support methodologies, the RBF network will learn over time and become increasingly accurate. During Phase II, this RBF network or an RBF network hybrid will be improved upon. The RBF triage decision support system was, therefore, a success.

Finally an overall architecture was designed. This architecture will allow for quick and easy expansion as new data, sensors, and program requirements are identified. The new requirements, such as diagnosis and treatment recommendations, will be able to be added to the system immediately upon its development. Therefore, the program can be deployed when the triage section is complete. Once the diagnostic capabilities are developed, they will be able to be added. This will prevent delays. The architecture proposed above will also deal with the possibility of new sensors, such as tissue pH, being added in the future. It will also handle a situation where a sensor is damaged on the battle field. As opposed to simply shutting down, the system will turn off the decision support systems requiring the information from the damaged sensor and turning on a backup routine which does not require the information. This architecture will also allow for any decision support system methodology such as neural networks, fuzzy logic, pattern recognition, or statistical analysis. Since any methodology at any time can be added to the system, this will allow the military to

contract out the various pieces for competitive bidding. This will ensure reliability at the lowest possible cost. This portion of the research was considered by the Principal Investigator to be successful.

Since the three areas of research in this Phase I SBIR were successful, this Phase I SBIR was considered to be an overall success. The PI of this Phase I SBIR will, therefore, propose a Phase II research proposal to be submitted before September 30, 1996.

REFERENCES

Books

Injury Severity Scoring, Current Therapy of Trauma, Third Edition, pp.8-16, B.C. Decker, Philadelphia, PA, 1991, Howard R. Champion, FRCS, FACS, Wayne Copes, Ph.D., William J. Sacco, Ph.D.

Trauma Scoring, Trauma, Second Edition, pp. 47-66, Appleton and Lange, Norwalk, Connecticut, 1991, Howard R. Champion, William J. Sacco, Wayne S. Copes.

Trauma, Mattox, et. al., Chapter on Trauma Scoring, Appleton & Lange, 1991.

Blunt Multiple Trauma; Marcel Dekker, Inc., 1990, pp. 261-276.

Trauma Registries, Advances in Trauma, Volume 3, 1988, p.241-262, Howard R. Champion, FRCS (Edin.), Wayne S. Copes, Ph.D., William J. Sacco, Ph.D.

Articles

Champion H., Yackee E., Gainer P.; A progress report on the Trauma Score in Predicting a Fatal Outcome; J. Trauma 26:10, 1986.

Champion H., Copes W., Sacco W., et. al.; The Major Trauma Outcome Study: Establishing National Norms for Trauma Care; J. Trauma 30:11, 1990.

Cios, K.J., Wedding, D.K., Serpen, G.; "RBF Networks with Fuzzy Covariance Matrices" *International Conference on Neural Networks (ICNN) '96, Washington, D.C. June 3-6, 1996*; Volume: Plenary, Panel, and Special Sessions, pp.187-192; 1996. (Invited Paper).

Sacco W., Thompson G., Copes W.; Development of Improved Personnel Serious and Lethal Models for Fragments and Burns; Survice-TR-95-010, Apr. 1995.

Savaia A., Moore F., Moore E., et al; Epidemiology of Trauma Deaths: A Reassessment.; J Trauma, 38:2; 1995.

Wedding, D.K. and Cios, K.J.; "Time Series Forecasting by Combining RBF Networks, Certainty Factors, and the Box-Jenkins Model" *Neurocomputing (10)* (special issue on financial applications), pp. 149-168; 1996.

Wedding, D.K. and Chick, D.E.; "NADEP NORIS Uses Alsys Ada to Develop Low Cost Trainer" *Alsynews*; Vol. 5 Number 3; Fall 1991.

Wedding, D.K., Cios, K.J., and Goodenday, L.S.; "A Bayesian Approach for Dealing with Uncertainties in Detection of Coronary Artery Stenosis Using a Knowledge-Based System" *IEEE Engineering in Medicine and Biology Magazine*; December, 1989.

APPENDIX

A company was located which puts P.C. motherboards onto small circuit boards which are slightly larger than a credit card. Thus, the concept of the soldier carrying a small computer is now commercially available. From preliminary discussions, it is the Principal Investigator's understanding that the company provides 386 and 486 computers and up to 32 MEG of RAM on the credit card computer. It is also the Principal Investigators understanding that a Pentium processor will soon be available. Although this information does not directly affect this SBIR, the information is related because the algorithm developed in this research will likely reside in a similar such computer. The company's name, address, and telephone number are included below.

S-MOS Systems
2460 North First Street
SanJose, CA 95131-1002
TEL 408-922-0200
FAX 408-922-0238

DEPARTMENT OF THE ARMY
US ARMY MEDICAL RESEARCH AND MATERIEL COMMAND
504 SCOTT STREET
FORT DETRICK, MARYLAND 21702-5012

REPLY TO
ATTENTION OF:

MCMR-RMI-S (70-1y) 4 Dec 02

MEMORANDUM FOR Administrator, Defense Technical Information Center (DTIC-OCA), 8725 John J. Kingman Road, Fort Belvoir, VA 22060-6218

SUBJECT: Request Change in Distribution Statement

1. The U.S. Army Medical Research and Materiel Command has reexamined the need for the limitation assigned to technical reports written for this Command. Request the limited distribution statement for the enclosed accession numbers be changed to "Approved for public release; distribution unlimited." These reports should be released to the National Technical Information Service.

2. Point of contact for this request is Ms. Kristin Morrow at DSN 343-7327 or by e-mail at Kristin.Morrow@det.amedd.army.mil.

FOR THE COMMANDER:

Phyllis Rinehart

Encl

PHYLLIS M. RINEHART
Deputy Chief of Staff for
Information Management

ADB218773
ADB223531
ADB230017
ADB223528
ADB231930
ADB226038
ADB224296
ADB228898
ADB216077
ADB218568
ADB216713
ADB216627
ADB215717
ADB218709
ADB216942
ADB216071
ADB215736
ADB216715
ADB215485
ADB215487
ADB220304
ADB215719
ADB216072
ADB222892
ADB215914
ADB222994
ADB216066
ADB217309
ADB216726
ADB216947
ADB227451
ADB229334
ADB228982
ADB227216
ADB224877
ADB224876
ADB227768
ADB228161
ADB229442
ADB230946
ADB230047
ADB225895
ADB229467
ADB224342
ADB230950
ADB227185
ADB231856

ADB229914
ADB229497
ADB230947
ADB282209
ADB270846
ADB282266
ADB262442
ADB256670